Spectrum Version 2 Addendum

This document describes the changes you'll find in Spectrum version 2. Spectrum version 2 contains fixes for several problems found in version 1, and it adds *many* new features and scripting capabilities.

Credits

Spectrum version 2 was a team effort that would not have been possible without all of our beta testers, so we thank all of them for their efforts to make Spectrum as bug free as possible.

We'd especially like to thank Ken Lucke for alpha testing Spectrum many months before it went into beta (his diligence in testing and retesting each incremental update resulted in a very smooth beta testing period). Thanks also for writing and maintaining the Help files, for all the scripting suggestions, and for writing scripts that actually *used* those new commands! Finally, thanks for the general support and encouragement; it has been appreciated.

And the greatest appreciation must go to Ewen Wannop who spent too many evenings and weekends writing version 2 instead of gardening! If it wasn't for his incredible patience in adding new scripting features at every turn, Spectrum wouldn't be what it is today.

Table Of Contents

Installation Instructions

If you just purchased the complete Spectrum version 2: Follow all the installation instructions in the *Getting Started and Reference* manual. If you install Spectrum sounds you should also run the "SoundPatch" utility to add some new sound events into the Sound CDEV.

To learn about most of Spectrum, read the *Getting Started and Reference* manual. Once you are familiar with the original Spectrum features you can then read the "New or Changed Program Features" section below to find out what's new in version 2. Likewise, when you begin writing scripts you should read the *Scripting* manual to learn about the original Spectrum features, then read the "Script Language Changes" section below to learn what's new in version 2.

If you purchased the update to Spectrum version 2:

① Insert the original Spectrum Update disk.

② Launch the "Prefs.Convert" application and follow the on-screen prompts (if any) to locate the old Spectrum version 1 and its preference file.

After copying and converting your old preferences into the new "Spectrum2.Prefs" file (inside the *:System:Preferences: folder) the conversion application will launch "Install' automatically.

③ When you are asked to do so, type your full name then click the OK button.

When personalization is complete, Apple's Installer is launched automatically.

If you use Spectrum on a hard disk drive:
Skip to the next page if you use Spectrum with 3.5" disks.

④ The "Update Spectrum" script is already highlighted, so just click the Install button and outdated version 1 files will be removed, then Spectrum v2 will be copied to the disk that was used to start up the computer. *NOTE: If you want to install Spectrum to some other disk, click the Disk button until the disk is displayed.*

⑤ After the update is installed, click the Quit button.

⑥ Launch the "SoundPatch" application and follow the on-screen prompts to update your Sound Control Panel (CDEV) so it lists *all* the currently defined "sound events."

⑦ Choose Control Panels from the (Apple) menu, then open the "Sound" CDEV.

⑧ Assign the sounds you want by selecting an event from the Event popup menu, then choosing the sound to be associated with that event from the Sound popup menu. For example:

Event	Suggested Sound
File Transfer Complete	Send/Receive-Good **or** Ahh **or** Trumpets
File Transfer Failed/Abort	Send/Receive-Bad
Grabbing Screen	Saving Screen
Key Click	Key-Any
Modem Connected	Phone-Connected
Modem Hanging Up	Phone-Hangup
Modem No Carrier	Phone-No Connection
RealTime Message	Phone-Connected **or** Trumpets
Return Key	Key-Return
Space Key	Key-Spacebar
You Have Mail	You Have Mail **or** You Have Mail (HAL)

⑨ If you experimented in step ⑧ by choosing several sounds to hear what they sound like, choose Shut Down from the Special menu, then select the Restart option and click OK to restart the computer. The reason for restarting is that the Sound CDEV is now probably using a *lot* more memory than it needs to be, and restarting will fix that.

⑩ Now launch Spectrum version 2!

If you use Spectrum with 3.5" disks:

See the previous page if you use Spectrum with a hard disk drive.

④ Choose Quit from the File menu and the Finder reappears.

⑤ Make a backup copy of the Spectrum update disk.

⑥ Drag the original Spectrum update disk icon into the Trash to eject the disk and remove its icon from the desktop.

⑦ Insert the backup copy of the Spectrum update disk.

⑧ Rename the backup disk to be "Spectrum".

⑨ Drag the following items into the Trash:

❑ Prefs.Convert
❑ Install
❑ Misc.

⑩ Choose Shut Down from the Special menu then click OK to shut down (turn off system power). You can now start using Spectrum v2 like you have been using Spectrum v1.

Notes

Bug Fixes

This section lists bugs in version 1 that were fixed in version 2. Bugs marked {INTERNAL} were discovered by Seven Hills Software; other bugs were reported by end-users.

General

- Install (the personalization program) was modified to work correctly with single 3.5" disk drive systems. We still *strongly* recommend using a hard disk drive (if you don't have one, get one...you won't regret it).

- When launched on some machines, Spectrum would "hang" until the CDA menu was accessed. Fixed.

- If HardPressed had to decompress Spectrum's resource fork when Spectrum was started, when Spectrum quit it would close the wrong resource file. Fixed (many thanks to Bill Tudor).

  Later, Spectrum was changed so it no longer opens its own resource fork for writing, which means you can compress Spectrum's data and resource forks if you wish.

- Fixed Spectrum's rBundle resource so the Finder would not keep adding it into the Desktop file. Also added all of Spectrum's custom icons into the rBundle (many thanks to Bill Tudor and Andy Wells) so the "classic" icon file is no longer needed.

- Spectrum's port driver has been improved significantly.

File Menu

Save

- Now text files remain text files, and Teach files remain Teach files.

- Fixed "black text," bogus memory errors, and other problems relating to bad Teach style information saved from the editor.

Send/Receive File

- When receiving a file that is saved automatically into the default file transfer folder, if the default folder is not found an error will be shown.
- If "Binary II Down" is checked and a file with a Binary II header is received, under some conditions the received file would be moved into the wrong folder. Fixed.
- {INTERNAL} Fixed a problem where sometimes checking the Prompted checkbox (when sending a text file) would still send the file unprompted.
- {INTERNAL} Uploading via Xmodem or Ymodem with the Binary II Up and ProDOS Xmodem options checked caused junk to appear at the end of the file. Fixed.

Capture

- Fixed a problem in the "auto save" name incrementing routine that could cause Spectrum to get stuck in an infinite loop trying to determine a unique filename.
- If the capture buffer size was set to "none" and you were capturing to a disk file, the Capture On/Off status was not being reflected in the thermometer that shows in the menu bar (if the Status Line option is checked). Fixed.
- {INTERNAL} = worked from Spectrum Text except when capturing to a file. Fixed.

Close Capture File

- If the capture file had been opened by a script, Close Capture File would result in a "bad pathname syntax" error dialog box. Fixed.

Print

- Apple's "Printer.Port" and "Modem.Port" port drivers install an interrupt routine when you print through that port driver, but the routine is not removed after printing is complete so incoming data might be lost. Unfortunately we can do nothing about this in Spectrum; it will require an update to (or replacement of) Apple's port drivers.

Edit Menu

- A bug was discovered in System 6 that causes Bad Things to happen when text has been copied to the Clipboard using a font that is no longer present in the system.

  This system bug appeared frequently with Spectrum because while using Spectrum's editor the default font was usually Spectrum.8, which is a font held internally to Spectrum (i.e. it did not exist in the Fonts folder).

  When you copied text to the Clipboard that was formatted with Spectrum.8, then quit Spectrum, the Clipboard would be corrupt and problems could occur.

  Unless/until Apple (or some third party) solves this problem, the best solution is for us to provide Spectrum.8 as an actual font that will reside in the *:System:Fonts folder (which we have done).

Show Menu

Scrollback

- Fixed a problem that could cause garbage to be inserted at the end of the Scrollback window.

Chat Line

- Any data that arrived while the chatline was being sent would not be processed through the correct filters. Fixed.

Phone Menu

Dial Number

- Fixed a problem that could have corrupted some memory when creating or editing a phonebook entry.
- Fixed a situation where the up/down arrows might start dialing the newly-highlighted phonebook entry.

Script Menu

Learn A Script

- Learning the "Load Menu File" menu item no longer repeats the filename.

Run A Script

- {INTERNAL} If a script was running and the Script menu was highlighted, pressing R would stop the script, but the next R would not present the Run A Script dialog box; you had to press R a third time. Fixed.

Rerun Script In Memory

- {INTERNAL} If you ran a script from disk, then pressed – when there was nothing in the editor, * would not rerun the script. Fixed (if the editor is empty then Run Editor As Script is dimmed).

Script Commands

- {INTERNAL} If **ScriptKeys** was On and a WaitFor statement was being executed, pressing Option-# would not "gosub" to the numbered label, and the WaitFor would never succeed. Fixed.
- {INTERNAL} The following script will now work:

```
Set LowASCII On; Set Quote $$A5
Display •one "two" three•
```

- {INTERNAL} There were some cases where a **Receive File** file transfer could fail but the Failed flag would not get set. Fixed.
- **Copy File** can now copy even very large ("tree") files, where previously it showed an "access not allowed" error.
- **Dial String** no longer reports a bogus error message before dialing.
- **Set ScreenBlank On** while the ANSI display is active will no longer result in a black screen with a yellow menu bar. Also, if Twilight II is installed Spectrum disables T2 and handles the screen blank itself.
- When **Open File** has to create a new text file, it now always sets the auxtype to $0000.
- Writing a text file then attempting to show it would fail with an "access not allowed" error. Fixed (**Show File** wasn't explicitly setting the open access to "read"; it does now).
- Incoming data now passes through the filters before **WaitFor String** checks for a match. This means the following script now works even if the host sends *one^M^Jtwo*:

```
Set StripLFs ON # the default situation
WaitFor String "one^Mtwo"; Display "Got it!^M^J"
```

- **Quit/Exit/Launch** now restores the key translation before quitting.
- **WaitFor Time** will now show times as "23:08" instead of "23: 8".
- **Read Catalog** was restricting the variable number to 0-3. Fixed; can now use any variable name.
- **Close CaptureFile** no longer results in a "bad pathname syntax" error dialog box.

- **If Even** / **If Not Even** / **If Odd** / **If Not Odd**: If the specified variable does not contain a valid number, or it contains the number 0 (zero), the Failed flag is set and control drops to the next line.

- MyLabel will now be found in the following script:

```
Goto MyLabel
# MyLabel  <=-There is a TAB right there
```

- The **Set FileXferPath** command did not work correctly when a ProDOS-8-style (slash delimited) path was used to specify a folder.

  Because Set FileXferPath is a Spectrum command you should always use Spectrum replacements and colon-delimited names (i.e. slash delimited names should appear only in scripts being converted from Talk Is Cheap). However, the Set FileXferPath command has been modified to work with slashes just in case.

- {INTERNAL} The **LowASCII** setting was not being honored for some commands, and could affect script interpretation in some cases. Fixed.

- {INTERNAL} The **CaseSensitive** setting could affect script interpretation and replacement item expansion. Fixed.

- {INTERNAL} **Send ScriptEditor** now honors the Screen flag.

- {INTERNAL} If **Send ScriptEditor** was used many times, the capture buffer would record a "successful" statement but the noted file size would keep increasing even if the same data had been sent each time. Fixed.

- {INTERNAL} Sometimes file number 0 would not be closed automatically when a script stopped. Fixed.

- Get Random was artificially restricting a random number to the maximum desired, resulting in the maximum number being hit more frequently than it should have been. Now improved so it is "more random."

Settings Menu

- A rare case where the Settings/Status dialog did not update the Editor File popup menu has been fixed.

- Resizing the Scrollback/Capture Buffer with existing data in the buffer often resulted in a bogus buffer. Fixed.

- Pressing H to Hangup no longer causes the mouse pointer to show on the ANSI display.

- A *much*-improved VT100 v1.1 display was available almost immediately after shipping Spectrum v1, but due to a production error it didn't get included on the Spectrum disk until recently. This Spectrum v2 upgrade includes version 2.0 of the VT100 display, which contains the major improvements found in v1.1, plus a few additional fixes:

  Now sends the correct response to the VT100 inquiry sequence, and fixed several other "bad responses."

  Fixed a problem where some partial control sequences could make it into the Capture/Scrollback buffers even though they were properly filtered from the screen.

Notes

New or Changed Program Features

This sections lists features that are new to version 2, or features which behave differently between versions 1 and 2. Most of these improvements resulted (directly or indirectly) from user feedback.

General

- The _HandleDiskInsert call was removed from Spectrum's main event loop because some disk drivers would mask interrupts when checking drive status. Now disk insertions will not be seen until you open a standard "open" or "save" file dialog box.
- If the Hardware Handshake checkbox is **not** checked then the system beep (_SysBeep) will be a buzzing sound. If it **is** checked then the normal system beep will sound.

  If you get a regular beep sound and incoming data is lost at that point, then you do not have a properly-wired modem cable (or your modem is not initialized correctly) so you should uncheck the Hardware Handshake option for every phonebook entry (select an entry, **click Edit,** then click Port Settings).
- When on a super hi-res screen, Shift-3 and the Save Screen script command now saves the screen picture *and* a text file that contains the contents of the screen. *NOTE: MouseText or other special characters will become regular text in the text file.*
- Shift-3 and Shift-4 now also work within the Editor, Capture Buffer, Scrollback, and Clipboard windows.
- Changed the way the preferences are stored in an attempt to prevent damage to the preference file if the system crashes. Also, passwords and character filters are now stored inside the preference file instead of Spectrum.
- Spectrum's port drivers (e.g. Spectrum.Port, SP.Port.Alt, etc.) are now completely responsible for putting data into, and getting data from, the port buffer. This means that updated port drivers can be released without modifying Spectrum.

- Spectrum now plays its sounds by event number, and the user can use the Sound Control Panel (CDEV) to select which sound is associated with each event (if he has run the “SoundPatch” utility).

- Added some support for extended keyboards:

  While an Online Display is frontmost:

F1-F15	attempts to “gosub” to a numbered label (see Set Labels in the Scripting manual).

  While the Editor window is frontmost:

F1	Undo
F2	Cut
F3	Copy
F4	Paste
Home	moves cursor to beginning of text
End	moves cursor to end of text
Page Up	moves cursor to beginning of line
Page Down	moves cursor to end of line

- X, C, V editing keys can now be used to Cut, Copy, Paste in all line edit boxes.

- The first time you launch Spectrum, a Spectrum2.Prefs file will be stored in the *:System:Preferences: folder. If you move that Spectrum2.Prefs file to be in the same folder as Spectrum, then Spectrum will access its preferences from there instead of from *:System:Preferences:.

- The Add.Ons folder now contains folders for extra Help.Files, Online.Displays, and XCMDs. The only files that should be in the Add.Ons folder itself are the Spectrum.Help and Spectrum.Port files.

- While launching, Spectrum displays messages as it loads XCMDs and Online Displays.

  NOTE: After the last XCMD is loaded, a message says “Please wait; starting XCMDs.” This message is shown because some XCMDs can take a long time to start. However, if all your installed XCMDs start quickly, this message will disappear before you can read it. That is normal.

External Commands (XCMDs)

Spectrum now supports "Spectrum External Commands." XCMDs can provide any number of new features, just by dropping a file into the XCMDs folder.

If you hold down the Shift key while Spectrum is loading, no XCMDs will be loaded. This is useful when investigating system conflicts, and on machines with low memory. *NOTE: You can inactivate an XCMD in the Finder by highlighting the XCMD (in the Spectrum:Add.Ons:XCMDs: folder) then choosing Icon Info and checking the "Inactive" checkbox.*

Several XCMDs have been provided with Spectrum; open the "Documentation: XCMDs" folder to read about them. *NOTE: If you are a programmer and want to write XCMDs for Spectrum, please contact Seven Hills Software.*

🍎 (Apple) Menu

Help

- Help text can now be highlighted and copied to the clipboard (⌘C).

File Menu

Save

- If the capture buffer fills and you opt to save it with the "keep appending" option checked, from then on choosing Save from the File menu will append the capture buffer to the file then clear the capture buffer.

- When saving a new Editor document, the default format will be "Text" unless a font, size, style, or color change was made (in which which case the default format will be "Teach").

- If a font, size, style, or color change was made to the Editor since last being saved, and you attempt to save as a Text file, a warning message is shown that all style information will be lost.

Send/Receive File

- When using a data format that is not 8-bit, the send/receive file menu items which require 8-bits of data are now dimmed to indicate you cannot use them.

  NOTE: If the host is capable of 8-bit file transfers then the host is an 8-bit system. This means you ***can*** *log in with an 8-bit data format; just be sure to check the "Convert to low ASCII option" in the Online Display Settings dialog box to ensure that text displayed on the screen is readable.*

- When resuming a Zmodem transfer, Spectrum previously asked for the first packet to determine which packet it really needed, then asked for the true packet. For some reason, GEnie recognizes only the *first* reposition request, so transfers would not resume correctly on GEnie.

  To work around this problem, Spectrum now appends resume information onto partially-received files, so the first reposition command will be to the packet it needs.

  Two bonuses: (1) This resume information also includes the Binary II setting, so Zmodem transfers will always resume properly even if you fiddle with the Binary II options between attempts (as long as the *host* does not change how it sends the files between attempts). (2) If a file already exists on disk, Spectrum will only resume to it if it is a partial file (if the file does not have the resume packet then it's a "complete" file and Spectrum picks a new name for the download).

- CIS B+ transfers would work only with ProDOS disks with Binary II Down turned off; now works with any disk format and will resume regardless of the Binary II Down setting (as long as the *host* does not change how it sends the files between attempts).

- You can now send text files as lines, and can even specify the line break lengths. After choosing a text file to send, the "send text settings" dialog box appears (which contains the same options as when sending the editor text).

- When receiving a batch of files, Spectrum now resets the timers and counters between each file, rather than showing an overall total. The CPS rate of the previous file is shown until the CPS rate for the current file stabilizes.

- Spectrum was going overboard in its confirmation checks after receiving a file:

  via Ymodem would sometimes take a long time to close the file transfer dialog; now it will close sooner

  via Zmodem would set the Failed flag if a confirmation packet timed out, even though the file itself actually transferred successfully

- During file transfers a custom "wait" mouse pointer is used. By not calling _WaitCursor, cursor animation programs (e.g. Cool Cursor, Wait Watch, etc.) will not attempt to animate the cursor during a file transfer.

  On a related note, GS+ Magazine is publishing an update to Cool Cursor, along with a "GSPlus" XCMD which provides several features:

  it automatically tells Cool Cursor not to animate the mouse pointer any time a carrier is detected in Spectrum

  it gives scripts direct control over Cool Cursor

  it gives scripts access to their "Quick DA" utility, so a script can open a CDA, NDA, or CDEV

  For more information (or to subscribe to their great magazine) contact:

GS+ Magazine	(615) 332-2087 *inquiries*
P.O. Box 15366	(800) 662-3634 *orders*
Chattanooga, TN 37415-0366	(615) 332-2634 *fax*

  America Online, Delphi: *GSPlusDiz*
  Internet: *gsplusdiz@aol.com*
  GEnie: *Wankerl*

  And while you're talking to them, purchase Auto Ark—even if it's just to get your hands on "Balloon" and the "Balloon XCMD" that comes bundled with it!

  Balloon is an NDA (and Finder Extra) that makes packing and unpacking files a breeze (imagine GS-ShrinkIt as an NDA). That's cool enough, but drop the Balloon XCMD into Spectrum's XCMDs folder and, like magic, the files you download will be *unpacked automatically* after you hang up!

Capture To File

- When starting a capture to a disk file, the Capture item is automatically checked to ensure that incoming data is actually captured.

  If the Capture option is turned on automatically, and you (or a script) do not manually turn the option off/on while capturing, then option will be turned back off when the capture file is closed.

Launch

- Choosing Launch from the File menu causes the Spectrum.Launch script to be run. If nothing needs to be saved before launching, you will be asked if you are sure you want to launch. This insures that you have at least one opportunity to cancel the process.

  NOTE: Because Spectrum.Launch is a script, advanced users can modify the script to perform other, custom actions when launching.

Quit

- Choosing Quit from the File menu causes the Spectrum.Quit script to be run. If nothing needs to be saved before quitting, you will be asked if you are sure you want to quit. This insures that you have at least one opportunity to cancel the process.

  NOTE: Because Spectrum.Quit is a script, advanced users can modify the script to perform other, custom actions when quitting.

Edit Menu

Paste / Send Clipboard To Modem

- Paste still always pastes the Clipboard as if you were typing it at the keyboard. So pasted text appears in the chatline if it is active, the editor if it is active, and so on.

  Send Clipboard to Modem used to do the same thing, but it now sends the Clipboard directly to the modem (as the name indicates). Also, this option respects the PadCR and Send LFs settings.

Style

- A new "Style" menu item has been added. Highlighting the Style item presents a sub-menu with the available text styles. This is a shortcut for applying styles in the Editor (avoids opening the Choose Font dialog box).

Format Text

- The "Convert Lines to Paragraph" option has been significantly improved.

Find/Replace

- The Find/Replace menu item is now active for the Scrollback and Capture buffer windows. When acting on the Scrollback or Capture Buffer window, the "Replace" and "Replace All" options are dimmed because they do not apply.

Show Menu

- Menu dimming was improved so that when the editor window was frontmost, the Editor menu item will be dim (likewise for the other items).

Capture Buffer

- Shift-⌘\ (or ⌘|) now opens the Capture Buffer window.

Scrollback

- A new script command (Set ScrollData) lets you change the way data is stored into the Scrollback buffer. You can either store the raw data (as in version 1) or you can store the filtered data (what is put into the capture buffer). The default setting is to store the filtered data.

Chat Line

- The chat line can be edited. Arrow keys position the cursor within the chat line, Delete removes characters, and typing inserts new characters. As before, Clear erases the entire chat line, and Return sends the entire chat line regardless of where the cursor is located.

Phone Menu

Dial Number

- When dialing, the Redial button is now disabled until after Spectrum has output the dial string to the modem.

- Phonebook entries now remember up to a 64-character pathname for the logon script. As in version 1, if only a name is entered then the script is run from the folder that contains the current menu file. The logon script path can be entered by hand, or by using the Select button.

- A specific Online Display can be remembered for each phonebook entry. "No change" does not change the current Online Display, while the other choices will open the specified Online Display immediately after a connection has been detected.

- To make direct connections easier you can now create a phonebook entry with no phone number. Clicking "Connect" establishes the Port and Online Display settings that are stored with the phonebook entry, then the logon script will be run (if one is attached to the phonebook entry).

- When dialing a phonebook entry, if dialing and redialing fails to make a connection, Spectrum automatically tries dialing the next phonebook entry.

Script Menu

This section covers general changes that script *users* might notice; changes for script *authors* are noted in the "Script Language Changes" section below.

Run A Script

- Scripts execute up to 300% faster than they did in version 1.
- When on an SHR display with the menu bar showing, the "script running" indicator (a black box in the menu bar) now shows only if the Status Line option is checked.
- Running a script no longer forces the Online Display to be open all the time. Now the display remains closed until a script command is encountered which requires the display to be open.
- All commands that require a *Statement* parameter now give an appropriate "parameter not assigned" error if a *Statement* is not provided. If a script you used with version 1 included a command such as...

  ```
  If Even 3; If Equal "$1" "A"; Display "Something"
  ```

  ...then you will see an error in version 2 because the required *Statement* parameters are not provided. The solution is to remove the semicolons:

  ```
  If Even 3 If Equal "$1" "A" Display "Something"
  ```

  Or, for better readability, include the optional "THEN":

  ```
  If Even 3 THEN If Equal "$1" "A" THEN Display "Something"
  ```

Run Editor As Script

- As in version 1, $ScriptFile is null ("") if a script is being run from the editor. What's new: If the editor has been saved, $ScriptPath will indicate the folder that contains the editor document. This lets you Chain to other scripts in the same folder as the script you're debugging in the editor. Also, if the script you Chain to includes the new Chainback command, the script in the editor will be resumed.

Settings Menu

Online Display

- If you press Option-Delete the *opposite* of the "Delete key = Backspace" setting will be used.
- At Spectrum startup time, Spectrum attempts to establish your desired buffers. In version 1, if there is not enough memory the buffer size would be set to zero, and you would have no clue this occurred.

  In version 2, Spectrum attempts to establish your desired Capture buffer and if there is not enough memory the buffer will be halved until successful. The process is then repeated for the Scrollback buffer.

  This method gives the Capture buffer higher priority, and lessens the chance of ending up with no capture buffer at all.

  Also, if the buffer sizes were not restored you now get a message to that effect, so you'll know to check them.

ANSI (v1.2)

- Fixed problem where the "More Display Options" settings sometimes would not be saved to disk.
- More Display Options dialog box was modified to better describe its features, and to add a new feature that lets you control whether color is supported at all (monochrome is easier to read, but not as cool).
- Now preserves special character attributes when closing and reopening the display.
- No longer captures control characters (except CR) or recognized ANSI control sequences.
- Fixed **ESC[1;J** to clear from beginning of line to current cursor position, inclusive (previously was clearing to end of screen).
- **ESC[nf** now does the same as "ESC[n;nH" (as per spec).
- **ESC[5n** now responds with **ESC[0n** to indicate "ready, no malfunctions detected."
- Added "ANSI-BBS" cursor keys.

- Noon now shows “12” instead of “0”.
- Although the display itself does not play ANSI music, it will broadcast music commands to an external module (if present) so that *it* may play the music.

ProTERM Special (v1.1)

- No longer captures control characters (except CR) or recognized PTSE control sequences.
- Now preserves special character attributes when closing and reopening the display.
- Fixed bug in DrawWindow which could draw garbage on the screen.
- Fixed a very rare bug where inverse/mousetext characters could get messed up.

Spectrum SHR Fast (v2.0) & Spectrum SHR Normal (v2.0)

- Both now show 23 lines with the chatline off; 19 with the chatline on.
- Both now support MouseText characters.
- Both now include the following DirectAction commands:

DirectAction "Page Mode On" *Varname*

Turns on the paged display mode (instead of scrolling the display, it will be wiped clean and the cursor put at 0,0). *Varname* is set to “OK”. This option dramatically speeds up the display of incoming text because no scrolling occurs, and is meant for use in situations where lots of text is being received that will be read offline later.

DirectAction "Page Mode Off" *Varname*

Turns off the paged display mode (when the screen needs to scroll, it will scroll one line at a time). *Varname* is set to “OK”.

DirectAction "TopMargin=?" *Varname*

Sets *Varname* to the current TopMargin value.

DirectAction "TopMargin=#" *Varname*

# is a value from 0-22 (chatline off) or 0-18 (chatline on)

Freezes the first # lines of the display. If an invalid number is used the Failed flag is set and *Varname* is cleared to "", otherwise *Varname* is set to “OK”.

The frozen area can be used by scripts to display messages, icons, etc. while incoming data is displayed in the area below the frozen area. For example:

```
Set Flush OFF
Display "^L"  # clear the whole screen
DirectAction "TopMargin=2" 0  # freeze the first 2 lines of
   the display
Display "^L"  # only affects the unfrozen part of the screen
Set Var 0 0  # set our counter to zero
# LOOP
Set Timeout 0; WaitFor String "^M"  # process incoming data
   until we see a Return character
INC 0  # increment our Return counter
STORE XY  # save the current cursor position
GotoXY 0,0; Display "CRs received so far: $0"
RESTORE XY  # restore previous cursor position
GOTO LOOP  # continue processing incoming data
```

Spectrum Text (v1.1)

- The “Press ^W to Close” message was removed to make room for messages that indicate the status of the capture buffer (whether on or off, and whether capturing to memory or to a disk file).
- Now preserves special character attributes when closing and reopening the display.

Viewdata (v1.3)

- Now has “smart number send” (click anywhere on a sequence of numbers and the entire sequence is transmitted, instead of just the character that was clicked on).
- The display shows the capture buffer contents so you can page through it.
- The Viewdata editor shows the contents of the Editor. To transfer data from the capture buffer to the editor you must Save it, open the Viewdata editor, then load the file (the same behavior as all other displays).

VT100 (v2.0)

- Now preserves special character attributes when closing and reopening the display.

...More Display Options

- As before, this menu item activates only after opening an Online Display that has additional display items. What's new is that simply opening the Online Display settings dialog box will no longer dim the item; it will dim only when a new Online Display is selected.

File Transfer

- The single "Auto Receive" checkbox has been replaced by two checkboxes:

  Auto B+ controls whether Spectrum automatically receives CompuServe B+ files.

  Auto Z controls whether Spectrum automatically receives Zmodem files.

  The default is to automatically receive Zmodem files (because the Zmodem trigger is a very special sequence that is not likely to occur within regular text) and to *not* automatically receive CompuServe B+ (because the trigger is a simple ^E).

Status

- The checkbox has been renamed to be more clear.

Notes

Script Language Changes

This section details new and changed script commands in version 2. It is intended for script *authors;* script *users* should find that scripts written for version 1 still execute identically under version 2 (just a whole lot faster). Script authors should also review the changes documented above, in the "Run A Script" section.

This section takes up more than half of this manual addendum, which gives you a clue that the scripting capabilities have been extended *tremendously*. A major leap forward is that instead of having only 10 variable numbers (0 through 9) Spectrum v2 supports any number of variable *names.*

There are many new commands that let you easily do sophisticated things, such as putting live "buttons" on the screen that can be clicked to perform some action.

In addition, Spectrum v2 supports "Spectrum External Commands" (or XCMDs). External commands provide sophisticated add-on features for scripts. For example, one XCMD lets a script present a standard IIGS window containing a "list box" from which the user can highlight items and click buttons; another XCMD gives scripts complete access to the IIGS' AlertWindow toolbox call.

And despite all these improvements and enhanced capabilities, we've managed to make scripts execute up to 3 times faster!

Specially-Treated Characters

$ (Replacement Items)

$FullTime

This now strips any trailing spaces (applies only to 24-hour time format).

$ErrorMsg

Some of the messages have changed for clarity. For example, "Parameter not assigned" is now "Required parameter is missing."

$ErrorCode

Only valid after a disk access error occurs

Gets replaced by the hexadecimal code for the disk access error that occurred. See the "Error.Codes" file in the Documentation folder for a list of possible errors.

$FilePos#

# must be replaced by a number from 0 to 9

Gets replaced by the character position of the "expected" file marker for the specified file number. If $FilePos3 is 7 then the next character that is read from file 3 will be the 7th character in the file; if a character is written then the 7th character in the file will be overwritten.

$ActualFilePos#

# must be replaced by a number from 0 to 9

Gets replaced by the actual file marker position for the specified file number. This value will match the $FilePos# value unless a Set FilePos command has been used (in which case $ActualFilePos# shows the *true* file position, while $FilePos# shows the *desired* file position).

$EOF#

# must be replaced by a number from 0 to 9

Gets replaced by the character position of the last character in the specified file (the End Of File marker).

$EditorSize#

# must be replaced by a number from 0 to 9

Gets replaced by the number of characters in the given script editor.

$EditorFile

Gets replaced by the name of the file in the editor (if the editor has been saved).

$LastScriptPath

Gets replaced by the *Foldername* of the script which chained to the currently-running script. If the currently-running script was not chained to, or was chained to from a script in the Editor, then *$LastScriptPath* is "".

$LastScriptFile

Gets replaced by the *Filename* of the script which chained to the currently-running script. If the currently-running script was not chained to, or was chained to from a script in the Editor, then *$LastScriptFile* is "".

$ReplyQuote

Gets replaced by the quote string (see Set ReplyQuote).

$Quote

Gets replaced with the current quote character (see Set Quote).

$Token

Gets replaced with the current token character (see Set Token).

$Baud

Gets replaced by the current baud rate, in the format used by the Set Baud command (i.e. 2400 or 9600).

$DFormat

Gets replaced by the current data format, in the format used by the Set DFormat command (i.e. 7E2 or 8N1).

$Duplex

Gets replaced by the current duplex setting, in the format used by the Set Duplex command (i.e. HALF or FULL).

$Echo

Gets replaced by the current echo setting, in the format used by the Set Echo command (i.e. OFF or ON).

$Buffer

Returns the current size of the capture buffer.

$StatLine

Gets replaced by OFF or ON to indicate the state of the status line.

$ChatLine

Gets replaced by OFF or ON to indicate the state of the chat line.

$SendLines

Gets replaced by the line break position, as set by the Set SendLines command.

$Hit

Gets replaced by the number of the last HitZone that was clicked.

$Update

If a screen update occurred on a super hires display, $Update is set to 1. This is useful for advanced scripts that may need to redraw special screen elements (rectangles, icons, etc.).

$Dialed

Intended for use in a logon script. Gets replaced by a null string if no phone number was dialed (i.e. the phonebook entry did not have a phone number attached). It is replaced by "Yes" if a phone number was dialed.

$PBMethod

Gets replaced by a "T" or "P" to indicate Tone or Pulse dialing, as set by the last-accessed phonebook entry.

$PBNumber

Gets replaced by the phone number of the last-accessed phonebook entry.

$PBDisplay

Gets replaced by the name of the Online Display that is attached to the last-accessed phonebook entry.

$PBRedial

Gets replaced by the redialing options set for the last-accessed phonebook entry.

$PBFile

Gets replaced by the *Filename* of the script (if any) that is attached to the last-accessed phonebook entry. (*$LogonFile* is still valid for compatibility with version 1.)

$PBPath

Gets replaced by the *Foldername* of the script (if any) that is attached to the last-accessed phonebook entry.

$SEDirty#

# must be replaced by a number from 0 to 9

Gets replaced by 1 if the specified script editor has been changed since being loaded or saved; null if the editor has not been changed.

$EditorDirty

Gets replaced by "1" if the editor contents need to be saved; otherwise it is null.

$BufferDirty

Gets replaced by "1" if the capture buffer needs to be saved; otherwise it is null.

$AutoSaveBuffer

Gets replaced by the state of the AutoSaveBuffer flag.

$EditorHandle#

# must be replaced by a number from 0 to 9

Gets replaced by the memory address of the TextEdit handle that holds the data for the specified script editor. For use when communicating with External commands.

$FailureCode

After a command that waits for port input, if Failed is true then $FailureCode gives the reason the command failed:

0 = Other source (e.g. a HitZone)
1 = Timeout was reached
2 = IdleTimer was reached

$DoHangup

After using the Get File command with a Kind of 3 (launchable applications), $DoHangup indicates whether the "Hangup" option was checked (1 means yes; null means no).

$DoReturn

After using the Get File command with a Kind of 3 (launchable applications), $DoReturn indicates whether the "Return to Spectrum" option was checked (1 means yes; null means no).

Parameters

VarNum becomes *Varname*

Spectrum now supports *named* variables—there is no longer any such thing as a "VarNum"! Every place you see "VarNum" in the script manual, substitute "Varname".

Spectrum still supports the ten numbered variables (0, 1, 2, 3, 4, 5, 6, 7, 8, and 9). With the exception of those default variables, variable names must start with a letter.

Variable names can contain any combination of letters and numbers, up to 32 characters long. Uppercase/lowercase does not matter (i.e., MYNUM, mynum, MyNum, and mYnUm are all the same variable).

The only naming restriction is that you cannot create a variable based on the name of a built-in replacement item. For example, you cannot use "Date" because it is a built-in replacement item, nor can you use variable names that begin with Date (e.g., "DateOfCall" or "Date2"). You may, however, use names such as "MyDate". *NOTE: An "Assigned name is reserved" script error occurs if you attempt to use an existing replacement name.*

Named variables work just like the numbered variables work: any place a *VarNum* was accepted, a *Varname* can now be used. The only minor difference between the two kinds of variables is that variables 0-9 are always defined and pre-set to null ("") by default, but named variables are not defined until they are created.

This means that when running a fresh script, displaying $0 will show nothing (because var 0 is pre-set to ""), but displaying $Something will show "$Something" on the screen because a variable named "Something" has not been created yet.

New named variables are created explicitly (e.g. Set Var MyNamedVar "Hi there!") or implicitly (e.g. Multiply 41 8 TheResult). Simply referring to a variable (i.e. using "$SomeVariable") does not create it.

In summary, *numbered* variables and *named* variables are nearly identical. The differences are:

- *Numbered* variables always exist; *Named* variables exist only after they are created.
- Certain commands (such as Store Vars, Restore Vars, Clear Vars, etc.) operate only on the 10 *numbered* variables.
- Certain commands (such as Delete Vars) work only on *named* variables.

Tips for Advanced Users

When setting a variable (Set Var MyColor "Blue") or accessing its contents ($MyColor) Spectrum must first find the exact variable being referenced before it can work with it.

Initially the variable name table is very short, but as you create new variables the table of variable names is expanded. The more variables you have, the longer the table is, and the longer it takes to search the table.

The search speed is not a concern unless you define hundreds of variables, but even then the search can be optimized by using shorter variable names, or long names that are unique in the first few characters.

For example, using variables named "My1VariableName" through "My99VariableName" will be faster than using variables named "MyVariableName01" through "MyVariableName99". And using "M1" through "M99" will be even faster.

If you anticipate defining many variables, consider combining several pieces of information into a single variable. For example, instead of having 100 different variables to remember a "yes/no" response you could use a single variable to remember each response (e.g. "YYNNNYYN...") and use the Substring/Overlay String commands to get and set the appropriate character.

Also, to keep the variable name table short, use the Delete Variables command to delete any variables created for temporary use.

String

A *String* can now contain up to 256 *Characters* (the previous limit was 128). Accordingly, this means that Variables, FKeys, Passwords, etc. can store up to 256 characters each.

Script Development

- Extra spaces, commas, and hard spaces no longer affect command recognition (e.g. "if contains" is now recognized as being the command "if contains").

Set Debug *State {"FoldernameFilename"}*

State can be External, File, Screen, Scrollback, or Off

FoldernameFilename is required when *State* is File, and must not be present for all other *States*

This command works like it did in version 1, with these additions:

- When Spectrum is executing a script returned from an XCMD, the debug statement shows "XCMD –=>" instead of "DEBUG –=>".

- "Set Debug External" broadcasts debugging information to Spectrum External Commands, in case there is an XCMD that wants to handle debugging in a custom way. A "Debug" XCMD is provided as an example to show what is possible.

- "Set Debug File" directs Scrollback data into the specified *Foldername-Filename* instead of to memory. This option significantly slows down script execution because the file is updated frequently. It is recommended the file be created on a RAM disk or a fast hard drive.

Example:

```
Set Debug File ":Ram5:Debug.File"
Display "Check the file when this quits."
AnUnknownCommandHere
```

Fundamental Commands

Play Sound "*Name*"

This command now plays any sound that can be seen by the resource manager, not just sounds stored in the *:System:Sounds: folder. *NOTE: Resource names are case sensitive.*

Play Event *Value*

Plays the sound for the specified event at the volume specified in the Sounds Control Panel. *NOTE: In order to play sounds the Sounds CDEV must be installed and active, and the "Sounds" checkbox must be on in the Online Display Settings dialog box.*

Example:

```
    Play Event $$8 # bad keypress
    Play Event $$100 # You have mail
```

The standard Sound CDEV does not list all these sound events. Run the "SoundPatch" utility to add listings for most of these events.

Sound Events	
0000 Alert stage 0	0053 Alert note
0001 Alert stage 1	0054 Alert caution
0002 Alert stage 2	0060 Screen blanking
0003 Alert stage 3	0061 Screen unblanking
0004 Outside window	0070 Beginning long operation
0005 Operation complete	0080 Application launching
0008 Bad keypress	0081 Application quitting
0009 Bad input value	0100 You have mail
000A Input field full	0E00 Error window base
000B Operation impossible	0EFF Error window other
000C Operation failed	0F00 Visual sound
0011 GSOS to P8	0F80 File transferred
0012 P8 to GSOS	0F81 Real time message
0013 Disk inserted	0F82 File transfer failed
0014 Disk ejected	1000 Connected to service
0015 System shut down	1001 Disconnected from service
0016 Volume changed	1002 Entered real time chat
0030 Disk request	1003 Left real time chat
0031 System startup	1004 Entering Forum/RT area
0032 System restart	1005 Leaving Forum/RT area
0033 Bad disk	1008 Modem-Dialing
0034 Key click	1009 Modem-Hanging up
0035 Return key	100A Modem-No carrier
0036 Space key	100B Modem-Connected
0040 Whoosh open	1010 Feature enabled
0041 Whoosh closed	1011 Feature disabled
0042 Fill trash	1012 Taking screen shot
0043 Empty trash	1020 Reading message
0050 Alert window	1021 Sending message
0052 Alert stop	

Settings

Save Settings " *FoldernameFilename*"

FoldernameFilename is optional

Saves Spectrum's current settings to disk If no file is specified, settings are saved to the current settings file. If a file is specified then settings are saved to that file, and it becomes the current settings file.

Load Settings " *FoldernameFilename*"

FoldernameFilename is optional

Loads Spectrum's current settings from disk. If no file is specified, settings are loaded from the current settings file. If a file is specified then settings are loaded from that file, and it becomes the current settings file.

Set ScrollData *Kind*

Kind can be Raw or Filtered

Determines how incoming data is stored into the Scrollback buffer. Raw stores the actual data that was received; Filtered stores data after it has been processed by the online display (i.e. the same information that appears in the Capture Buffer).

Set ReplyQuote " *String*"

String can be 1-16 characters

Sets the quote format used when choosing Paste As Reply from the Edit menu.

Online Display Settings

Close OnlineDisplay

Shortcut: **Offline**

Closes the current online display and displays Spectrum's 640 mode desktop.

IMPORTANT: This command no longer causes the script to stop (because scripts can now operate even when the Online Display is not open). For full compatibility with version 1, edit any version 1 scripts so that "Close OnlineDisplay" is followed by the "Stop Script" command.

Clear Desktop

Closes all Spectrum windows and NDAs, displaying Spectrum's 640 mode desktop.

File Transfer Settings

Set AutoReceive *State*

State can be Off, On, BPlus, or Zmodem

Controls the "Auto B+" and "Auto Z" checkboxes.

Off: Turns off both checkboxes.

On: Turns on both checkboxes.

BPlus: Turns on the BPlus checkbox.

Zmodem: Turns on the Zmodem checkbox.

Set BinaryII *State*

State can be Downloads, Uploads, Off, or On

Works as described in the Scripting manual, except Downloads and On will no longer uncheck the "Resume Transfers" checkbox (because resume now works regardless of the receiver's BinaryII setting).

Set SendLines *State*

State can be Off or On, or a number from 10-65535

Controls how text files and ScriptEditors are sent. Off sends files as-is; On splits paragraphs into lines of approximately 73 characters each; a Value splits paragraphs into lines of approximately the specified number of characters.

Dialing

Dial Service "*PhonebookEntry*"

Instead of generating a script error, now the Failed flag is set if the specified phonebook entry does not exist.

Dial Entry *Value*

Instead of generating a script error, now the Failed flag is set if the specified phonebook entry does not exist.

Get ServiceInfo "*PhonebookEntry*"

Establishes the port settings for the specified phonebook entry, and updates all the *$PB_* replacement items so they reflect the information stored in the phonebook entry.

The Failed flag is set if the specified phonebook entry does not exist.

Connected

Scripts should issue this command any time they establish a connection with another computer or modem.

Onhook

Scripts should issue this command any time they close a connection (or when they know a connection has been broken) with another computer or modem.

Script and Program Control

Run "*FoldernameFilename*" *Label*

Label is optional

Resets many script parameters then runs the specified script. If *Label* is given control passes to the specified label in the new script. If the label does not exist then a "label not found" error occurs.

Chain "*FoldernameFilename*" *Label*

Label is optional

Passes control (or "Chains") to the specified script, preserving all the settings that are normally reset when a script is Run.

If *Label* is given then control is passed to the specified label in the new script. If the label does not exist then a "label not found" error occurs.

REMINDER: Chaining to a script does not preserve the current For/Next loops, nor does it preserve the "Gosub" stack.

ChainBack

Loads the script that chained to the currently-running script (*$LastScriptPath $LastScriptFile*) and resumes executing script commands at the statement after the Chain command.

If the currently-running script was not chained to, then ChainBack stops the script.

Launch / Exit / Quit

These commands now set prefix 0 and 8 to the folder of the application being launched, which resolves path problems when launching some ProDOS 8 programs.

Quit2

This command works exactly like the "Quit" command, except it does not hangup first.

Variables

Store Variables

Shortcut: **Store Vars**

Stores the current values of the ten numbered variables (0, 1, 2, 3, 4, 5, 6, 7, 8, and 9). It does not store any named variables.

Restore Variables

Shortcut: **Restore Vars**

Restores the ten numbered variables (0, 1, 2, 3, 4, 5, 6, 7, 8, and 9) to the previously-stored values. It has no effect on any named variables.

Clear Variables *Varname1*, *Varname2* ... *Varname#*

Shortcut: **Clear Vars**
Varnames are optional

Clears the specified variable names to "". If no variable names are specified, the ten numbered variables (0, 1, 2, 3, 4, 5, 6, 7, 8, and 9) are cleared to "".

Delete Variables *Varname1*, *Varname2* ... *Varname#*

Shortcut: **Delete Vars** *Varname1*, *Varname2* ... *Varname#*

Deletes the specified variable name(s) from the internal table of variable names.

Match String "*TheLine*" "*String1*" ... "*String#*"

Looks to see if *String1* exists anywhere within *TheLine*. If so then $Matched is set to "1", $MatchString is set to *String1*, and the Failed flag is cleared.

If *String1* is not found within *TheLine* then the other strings are checked (the only limit on the number of strings is how many can fit into the script expansion buffer).

If no match is found then $Matched is set to "0", $MatchString is set to "", and the Failed flag is set.

Strip Spaces "*TheString*" *Varname*

Deletes all space characters from *TheString* and stores the results in *Varname.*

Trim Spaces " *TheString*" *Varname*

Deletes only the leading and trailing spaces from *TheString* and stores the results in *Varname.*

Overlay String "*String1*" "*String2*" *Start* *Varname*

Start can be from 1 to 256

Overlays *String2* on top of *String1*, starting at character position *Start*, and stores the result string into the specified *Varname*. The Failed flag is set if *String2* cannot be overlaid into the existing space used by *String1*.

For example:

```
Overlay String "12345" "ABC" 1 9
# var 9 is now "ABC45"
Overlay String "12345" "ABC" 3 9
# var 9 is now "12ABC"
Overlay String "12345" "ABC" 4 9
# the Failed flag is set because "ABC" cannot be overlaid
  onto "12345" starting at position 4 (var 9 is set to
  "12345")
```

Insert String "*Original*" "*New*" *Position* *Varname*

Position can be from 1 to 256

Inserts the *New* string into the *Original* string at character *Position*. If the combined string exceeds 256 characters the result is truncated to 256 characters and the Failed flag is set.

Example:

```
# Loop
Display "Insert where? "; Input Line Position
Insert String "****" "NEW" $Position Result
Display "Result is '$Result'.^M"; Goto Loop
# the results:
0 - script error; run the script again
1 - NEW****
2 - *NEW***
3 - **NEW**
4 - ***NEW*
5 - ****NEW
6 through 256 - ****NEW
257 or higher - script error
```

Delete String " *TheString*" *Start* *Length* *Varname*

Start can be from 1 to 256
Length can be from 1 to 256

Deletes *Length* characters from *TheString*, starting at position *Start*. If *Start* is 0 or *Start* is greater than the length of *TheString* the Failed flag is set and *Varname* is set to *TheString*.

Example:

```
Delete String "12345" 1 3 9
# var 9's now "45"
Delete String "12345" 3 3 9
# var 9's now "12"
Delete String "12345" 3 256 9
# var 9's now "12"
```

Evaluate " *Expression*" *Varname*

Evaluates the given string expression and sets *Varname* to the result. Valid operators are ( ) + – * / ^ ÷ and – (the last two operators are Option-/ and Option-Minus, respectively). *NOTE: To use the "power of" operator (^) you must type it twice (^^).*

The calculations are made on whole Integer numbers, so any remainders from divisions are lost, with the number being rounded down.

The Failed flag is set if *Expression* is not a valid math expression (i.e. there are characters other than numbers, operators or spaces, the parentheses are not balanced, etc.). The Failed flag is also set if any intermediate calculation overflows 32 bits (4,294,967,296), or if parsing the expression overflows the internal 256-byte parsing buffer.

Finally, the Failed flag is set if the result is a negative number, because negative numbers are not valid elsewhere in Spectrum's scripting language. However, in this case the result *Varname* will contain the positive answer (i.e. if an expression is calculated to be –5 then the Failed flag will be set and *Varname* will be set to 5).

Example:

```
Evaluate "2+(3^^4)-2+(3^^2)-3*4+(6/3+(12/4))" Result
Display "The result is $Result.^M"
Evaluate "2^^$Day" Result
Display "The result is $Result.^M"
```

Getting Input

Set IdleTimer *Value*

Value can be from 0 to 65535

When waiting for input using any "Getting Input" command that gets data from the port, Spectrum monitors the port activity. If the port is idle for *Value* seconds then the command will fail and Spectrum continues executing the script.

This command works in *conjunction* with the Timeout value (Timeout fails the command after *Value* seconds **regardless of port activity**, whereas IdleTimer fails the command after *Value* seconds of **no port activity**).

Example:

```
Set Timeout 120 # 2 minutes
Set IdleTimer 30 # 30 seconds
Transmit "Send me the entire file list!^M"
WaitFor String "END OF LIST"; If NOT Failed Then Display "The
   entire list was received within 2 minutes."; Stop Script
# otherwise the WaitFor command failed...
On $FailureCode GotoNext TimedOut, IdledOut
# if here then $FailureCode is 0...
Display "Failed for some other reason (a HitZone was probably
   clicked)."; Stop Script
# TimedOut - after 2 minutes there's still port activity
Display "Still receiving the list."; Stop Script
# IdledOut - gone 30 seconds with no port activity
Display "Either the list is complete but something happened
   that the 'end of list' string was not seen, OR the host is
   disconnected/lost in space."; Stop Script
```

Read Char *Varname*

Accepts one character from the port and stores it into the specified variable.

If a timeout was used and time ran out, the Failed flag is set.

Read Line *Varname*

Accepts up to 256 characters from the port. When a Return character is received the line is stored into the specified variable.

If a timeout was used and time ran out, the Failed flag is set and the specified variable contains the data that was read so far.

Ask1 "Question" *"Button1"* *Varname*
Ask2 "Question" *"Button1"* *"Button2"* *Varname*
Ask3 "Question" *"Button1"* *"Button2"* *"Button3"* *Varname*

Instead of the question being restricted to 68 characters long, these commands now accept questions up to 255 characters long (the size of the alert window is adjusted automatically so the message will fit).

Branching and Loops

IMPORTANT: The script keyword "GotoNext" cannot be separated as "Goto Next" (version 1 was more forgiving of this error; version 2 will instead attempt to GOTO a label named "Next" which could cause an error in some version 1 scripts).

Set Labels *State*

State can be Off or On

Works as before, but adds support for an extended keyboard: Pressing function keys 1 through 15 will attempt to "Gosub" to that label.

On *Value* Goto *Label1*, *Label2* ... *Label#*
On *Value* GotoNext *Label1*, *Label2* ... *Label#*
On *Value* Gosub *Label1*, *Label2* ... *Label#*
On *Value* GosubNext *Label1*, *Label2* ... *Label#*

These commands have been modified to allow as many labels as will fit in the script line buffer (currently 636 characters, minus the number of characters used for the command itself).

If Value is 0 or is greater than the number of labels supplied, the Failed flag is set and control continues to the next statement. For example, the Display statement would be executed in this script:

```
    On 5 Goto One, Two; Display "Didn't go anywhere"; Stop Script
```

For *LoopNumber Start Stop Step*

Works like version 1, except if the initial *Start* value is less than the initial *Stop* value the loop counts *backwards* by *Step* numbers.

Example:

```
    Display "1 to 100 by twos:^M^J"
    For 0 1 100 2; Display "$ForValue0^I"; Next 0
    Display "^M^J^M^J100 to 1 by twos:^M^J"
    For 0 100 1 2; Display "$ForValue0^I"; Next 0
```

On Compare *Value1 Value2* **GoXXX** *LessLabel*, *EqualLabel*, *MoreLabel*

GoXXX can be Goto, GotoNext, Gosub, or GosubNext

This command is similar to the Compare command, except that the result can be acted on immediately (an intermediate variable to hold the result is not required):

If *Value1* < *Value2* then control passes to *LessLabel*

If *Value1* = *Value2* then control passes to *EqualLabel*

If *Value1* > *Value2* then control passes to *MoreLabel*

Conditional Tests

If Not Contains "*ShortString*" "*LongString*" **Then** *Statement*

If *ShortString* is longer than *LongString* then *Statement* is now executed.

If Null *Varname* **Then** *Statement*
If Null "*String*" **Then** *Statement*
If Null *$EditorHandle#* **Then** *Statement*

If a variable name is supplied and the variable is empty ("") then *Statement* is executed.

If a string is supplied (the current quote delimiters are required) and the string is empty ("") then *Statement* is executed.

If an $EditorHandle# replacement item is supplied and the specified editor is empty ($EditorSize# is 0) then *Statement* is executed.

If Not Null *Varname* **Then** *Statement*
If Not Null "*String*" **Then** *Statement*
If Null *$EditorHandle#* **Then** *Statement*

If a variable name is supplied and the variable is **not** empty then *Statement* is executed.

If a string is supplied (the current quote delimiters are required) and the string is **not** empty then *Statement* is executed.

If an $EditorHandle# replacement item is supplied and the specified editor is **not** empty then *Statement* is executed.

If Defined *Varname* **Then** *Statement*

If the specified variable name has been defined then *Statement* is executed.

If Not Defined *Varname* Then *Statement*

If the specified variable name has **not** been defined then *Statement* is executed.

If Desktop Then *Statement*

If the 640 mode SHR display with menu bar is showing then *Statement* is executed.

If Not Desktop Then *Statement*

If the 640 mode SHR display with menu bar is **not** showing then *Statement* is executed.

Screen Appearance

Set Flush

If Flush was On in version 1, processing incoming data had greater priority over executing the script commands. In version 2 both functions are approximately equal priority, so script commands will execute regularly even if lots of incoming data is pending.

Set ScreenBlank *State*

State can be Off, On, or Auto

Spectrum now completely controls the blanking (previously if Twilight II was active Spectrum would ask it to blank the screen). Also, the border color is now set to black when blanked.

Set ScreenBlank OFF now just broadcasts the "systemSaysForceUndim" IPC message, so if Twilight II (for example) blanks in the background, your script can include a Set ScreenBlank OFF command to force the screen to be unblanked. *NOTE: It **is** acceptable to issue "Set Screen OFF" even if the screen is already unblanked.*

Capture Buffer Control

Append CaptureFile "*FoldernameFilename*"

Opens the specified text file and begins capturing incoming data to the end of it.

What's new: If the specified file does not exist it will be created automatically. If the specified file is not a text file then a script error is generated.

Set AutoSave "*FoldernameFilename*"

If AutoSave is specified, and AutoSaveBuffer is on, and Append is on, whenever the capture buffer fills it will be automatically appended to the AutoSave folder and filename.

In version 1, if the AutoSave folder or filename was invalid, Spectrum would not save the buffer. In version 2 Spectrum gives the user the opportunity to manually Clear, Save, or Append it.

OS Utilities

Copy File "*Foldername1Filename1*" "*Foldername2Filename2*"

Can now copy even very large ("tree") files, where previously it showed an "access not allowed" error.

Get FileInfo "*FoldernameFilename*" *Varname*

Works as documented, with the following additions to the returned string:

Start	Length	Information
59	6	Creation date in the format YYMMDD
65	4	Creation time in the format HHMM
69	6	Modification date in the format YYMMDD
75	4	Modification time in the format HHMM

Show File "*FoldernameFilename*" *Start Stop*

Start and *Stop* are optional

As in version 1, this command displays the file to the screen (and outputs it to the port if the Echo option is on). What's new:

- The optional *Start* and *Stop* values let you show a specific portion of the file.
- While a file is being shown you can press the Option or ⌘ key to pause the display.

Get File "*PromptString*" *Kind Varname*

Works as it did in version 1, but Kind can now also be a 3, which presents a Get File dialog box that displays launchable applications.

When using the *Kind* of 3 the dialog box also contains two checkboxes that let the user control whether the script should hang up before launching ($DoHangup), and whether quitting the next application should return to Spectrum ($DoReturn).

Reading and Writing Files

NOTE: Scripts can now open 10 files and 10 catalogs (previous limits were 4 files and 4 catalogs).

Read File *FileNumber Varname NumBytes*

FileNumber can be from 0 to 9
NumBytes is optional; if used it is a value from 1-256

If the optional *NumBytes* parameter is not provided then this command behaves exactly as it did in version 1.

If *NumBytes* is specified then this command reads exactly that many bytes from the file (it does not check for Return). If reading *NumBytes* passes the end of the file, the Failed flag is set and the variable contains the bytes that were read successfully.

Write File *FileNumber "String" NumBytes*

FileNumber can be from 0 to 9
NumBytes is optional; if used it is a value from 1-256

If *NumBytes* is not specified then all the characters of the *String* are written; if *NumBytes* is specified then only the first *NumBytes* characters of the *String* are written.

Set FilePos *FileNumber Position*

FileNumber can be from 0 to 9

After opening a file you can use this command to quickly move the file marker to a specific character position. For example, **Set FilePos 0 83** will set the file marker to the 83rd character in file 0.

If you set to a character position that is past the end of a file, the file length will not get extended until you begin writing to that position.

For example, this script...

```
Open File 0 ":Ram5:NewFile"
Set FilePos 0 10000
Set FilePos 0 10; Write File 0 "Word.^M"
Close File 0
```

...will result in a file that is only 15 characters long, not 10,000. To extend a file Spectrum writes as many Return characters as necessary (in the example above, Return characters are written to position 1 through 9, so the "W" of "Word" will start at the character position that was set).

Search File *FileNumber "String"*

FileNumber can be from 0 to 9

Searches the given file for the given string. The search starts at the current file position and proceeds toward the end of the file.

If *String* is found then the Failed flag is cleared and the file marker is set to the first character *after* the string. If *String* is not found then the Failed flag is set.

Reading Catalogs

Read Catalog *CatalogNumber Varname*

The information that is returned has been extended—see "Get FileInfo" in the OS Utilities section.

Script Editor

All script editor commands have been modified to allow an optional *EditorNumber*, a value from 0-9. If *EditorNumber* is not present, editor number 0 is assumed:
Clear ScriptEditor *EditorNumber*
Send ScriptEditor *EditorNumber*
Print ScriptEditor *EditorNumber*
Apply Replace *EditorNumber " FindString" " ReplaceString" Varname*
Apply LowASCII *EditorNumber*
Apply RemoveControls *EditorNumber*
Apply LFsToCRs *EditorNumber*
Apply RemoveSpaces *EditorNumber*
Apply Special *EditorNumber Value*
Apply Format *EditorNumber Value*

Append ScriptEditor *EditorNumber "Item" Kind*
Load ScriptEditor *EditorNumber " Item" Kind*

EditorNumber is optional; if used it is a value 0-9
Item can be a FoldernameFilename, or can be ::Scrollback, ::Buffer, or ::Clipboard
Kind is an optional Varname

Loads the specified data into the given ScriptEditor (Load replaces the script editor contents, while Append adds the new data to the end of the specified script editor).

If *Item* is a file on disk, an optional *Kind* varname can be used which will be set to indicate what kind of file was loaded: 1=Text, 2=Teach, 3=AppleWorks Classic.

Save ScriptEditor *EditorNumber "FoldernameFilename" Kind*

EditorNumber is optional; if used it is a value 0-9
Kind is optional; 1 saves as a Text file; 2 saves as a Teach file; 3 saves as an AppleWorks Classic file

Saves the specified scripteditor to disk. If *Kind* is not specified then the format is Text (if a text or AppleWorks Classic file was loaded) or Teach (if a Teach file was loaded). Otherwise the *Kind* value forces a specific file type to be saved (only Teach saves any style information; the other formats only save the text from the specified script editor).

Create ScriptEditor *EditorNumber*

Creates an empty scripteditor. If the scripteditor is already in use then the Failed flag is set and the existing scripteditor is not changed.

Show ScriptEditor *EditorNumber*

Displays the specified scripteditor on the screen. Holding down the Option key pauses the display; pressing Escape cancels the display.

Clear SEDirty *EditorNumber*

Clears the "dirty" flag for the specified editor.

Error Control

ResumeNext

Use only in an "On Error Goto" procedure

Continues running the script *after* the command that caused the error. *NOTE: This can be confusing and is intended for advanced script authors only.*

Script Interpretation

Set Quote *Character*

A script error could occur when the quote character was set to something other than the standard character (") and a line contained a semicolon:

```
    Set Quote %
    Display %This is fine.%
    Display %This ; failed.%
```

This has been fixed (i.e. the second Display command will no longer fail).

Advanced Commands

Open ResFile *FileNum* " *FoldernameFilename* "
FileNum can be from 0 to 9

Opens the resource fork of the specified file, which adds it to Spectrum's resource search path. The Failed flag is set if the specified file's resource fork could not be opened.

If a script frequently uses commands that access resources out of a particular file (such as Draw Picture and Draw Icon) the script can be sped up by opening the resource file in advance. This is much faster because Spectrum doesn't have to open and close the resource file around each command; Spectrum will instead just always find the resource in memory (the most-recently opened resource file is searched first).

This command lets scripts create and store all its related resources in one file. However, because these resources are opened and installed at the head of Spectrum's resource search path, script authors who use this feature must take care not to use resource ID numbers that conflict with any of Spectrum's resources (otherwise Strange Things may happen).

When creating a resource file for use with Spectrum, use resource IDs in the range $7001 and above to avoid conflicts. *NOTE: Resource names are case sensitive.*

Close ResFile *FileNum*
FileNum can be from 0 to 9

Closes the specified resource file.

Close ResFiles

Closes all resource files opened with the Open ResFile command.

Set Failed

Sets the Failed flag.

Clear Failed

Clears the Failed flag.

Shut Down

Closes any files opened by the script, calls the AutoSaveBuffer routine, stops the script, then shuts down the system.

Set DiskErrors *State*

State can be Off or On

When a disk-related error occurs (e.g. "file busy" or "disk full") Spectrum normally generates a script error. The script error can be caught and handled by the script using an On Error Goto routine.

Another approach is to turn DiskErrors off, which will not generate a script error for disk-related problems. Instead, when a command that accesses the disk is used, the Failed flag is set or cleared to indicate whether the command was successful.

If the disk command fails a script can use the $ErrorCode and $ErrorMsg replacement items to determine what error occurred.

NOTE: Only disk access commands are affected; actual script errors (e.g., using a file number that is already in use) will still cause a script error to appear.

Set ScriptLock *State*

State can be Off or On

ScriptLock can be turned on to prevent a script from being interrupted by the user. When ScriptLock is on:

- On the File menu: the Launch and Quit menu items are dimmed, and the Close menu item is dimmed when the Online Display is frontmost.
- On the Phone menu: the Answer Back menu item is dimmed.
- The entire Script menu is dimmed.
- Pressing Escape will not stop the script.

If you use this command you must be very careful that you provide some way to exit your script, otherwise the user will have to restart the computer to quit Spectrum!

Compress Script "*Source*" " *Destination*"

Source is the *FoldernameFilename* of a Spectrum Script file
Destination is a *FoldernameFilename* parameter

Compresses the data fork of the *Source* file and saves it as *Destination* with the "Compressed Spectrum Script" file/auxtype. A script error occurs if an error occurs.

This command is intended only to compress script files, so it works only on Teach or Text files whose data forks are less than 64K.

There is no corresponding Uncompress command—the only way to use a compressed script file is by choosing Run A Script from the Script menu, by running a script that RUNs or CHAINs to the script, or by double-clicking a compressed script in the Finder.

When compressing a script you should be sure to keep the *Source* file intact. To help prevent mistakes, *Destination* must not already exist, otherwise the Failed flag is set.

Save Editor

If the editor contents need to be saved, this command saves the contents. If the editor had not be saved before (i.e. it is a new file) then the command presents the standard file dialog box so the user can specify where to put the file and what to name it.

The Failed flag is set or cleared to indicate the success of the command.

External *CommandName Data*

Shortcut: **Ext**
CommandName is required
Data is optional (depends on the particular external)

This broadcasts an IPC message to `Spectrum XCMD~COMMANDNAME~`. The external acts on the command (what the command does, and the format of any *Data* being passed, is entirely up to the external module).

The Failed flag is cleared or set to indicate the success or failure of the command.

Specialty Commands

Set Update

Sets the $Update replacement item to 1. If an advanced script is watching the $Update replacement to see if a screen update is needed, this will make the script think Spectrum has just redrawn the display in response to a system update event.

Clear Update

Sets the $Update replacement item to 0. When a screen update occurs on a super hires display, $Update is set to 1, which is a signal to advanced scripts to redraw special screen elements (rectangles, icons, etc.). After the script updates the screen it would include a Clear Update command so the screen won't be updated again until another update is needed.

Draw Rectangle *Left Right Top Bottom Fill Frame Radius*

Works as it did in version 1, with the addition of the optional *Radius* value (1-64) to create rectangles with rounded corners.

Draw Icon *X,Y "Icon" "FoldernameFilename"*

Works as it did in version 1, except if the specified resource is not found the Failed flag is set (no script error).

Draw DimIcon *X,Y "Icon" "FoldernameFilename"*

This command is identical to the Draw Icon command, except that it draws the icon so it looks "dimmed."

Draw Picture *X,Y "Picture" "FoldernameFilename"*

Works as it did in version 1, except if the specified resource is not found the Failed flag is set (no script error).

Draw Text *X,Y "String" "Family" Size Style Color*

X is a value from 0 to 639
Y is a value from 0 to 186
Family is optional; if used it is a either value or a string that identifies the font
Size is optional; if used it is a value from 2 to 255
Style is optional; if used it is a value from 0 to 31
Color is optional; if used it is a value from 0 to 15

Draws *String* on the screen at coordinates (*X*, *Y*). The standard font is Shaston, 8 point, plain text, in black. If *Family* is specified then that font is used (if available in the system). If *Size* is specified then that size is used, and so on.

To determine the correct *Style* value, add the values for each style you want to use:

Plain = 0	Bold = 1	Italic = 2
Underline = 4	Outline = 8	Shadow = 16

For example, to use Bold+Italic+Shadow the *Style* value would be 19 (1+2+16).

Define HitZone *Num Left, Right, Top, Bottom CornerRadius Color*

Num is a value from 1 to 32
Left and *Right* are values from 0 to 640
Top and *Bottom* are values from 0 to 186
CornerRadius is optional; if used it is a number from 0-32
Color is optional; if used it is a value from 0 to 15

Defines a rectangular area that can act like a "button." *NOTE: This command works only on 640 mode super-hires screen displays.*

When hitzones are turned on, clicking inside a hitzone will highlight it. If the mouse button is released while inside a hitzone the $Hit replacement item will indicate which hitzone number was clicked.

Clear Hit

If $Hit is not zero then a hitzone was clicked. After performing the desired action your script must include a Clear Hit command to reset $Hit to 0 (otherwise your script will think that the user has clicked a hitzone again).

Delete HitZone *Num*

Num is a value from 1 to 32

Erases the frame around the specified hitzone, then deletes the hitzone so it is no longer active.

Delete HitZones

Deletes all the hitzones, but does not erase each individual hitzone (Display "^L" to clear the screen).

Set HitZones *State*

State can be Off or On

Determines whether hitzones are active or not (when Off, clicking inside a hitzone does nothing).

Set HitAction *Action*

Action is a value from 0-4

After the mouse button goes down in a hitzone the zone is highlighted and tracked. If the mouse button is released while still inside the hitzone then Spectrum sets the $Hit replacement item to the hitzone number that was clicked. Optionally, an additional action can occur:

Action	Result
0	Do nothing else
1	Post an ESCape keypress (will trigger on "On Escape" handler, or will abort the script if On Escape has not been used)
2	Post a Return keypress (will complete a Get Key, Get Line, Input Key, Input Line, or WaitFor Keyboard command)
3	Post a Return into the port input stream (will complete a Get Key, Get Line, Read Char, or Read Line command)
4	Post "HitZone" text into the port input stream (will complete a WaitFor String "HitZone" command)

Store HitZones

Stores the current hitzones and the associated settings.

Restore HitZones

Restores the previously-stored hitzones and settings and redraws the hitzones on the screen.

Set Hit *Num*

Num is a value from 0 to 65535

Sets the $Hit replacement to the specified number. This is mainly available for XCMDs to use, but could be useful if your script wants to "fake" a hit.

Set Cursor *Value*

Value is a value 0-5

Sets the mouse cursor to one of the following pointers:

#	Cursor Action
0	Hide (hides the current cursor)
1	Show (shows the current cursor)
2	Arrow
3	Watch (animates if a cursor animation extra is installed)
4	Spectrum Watch (won't animate)
5	I-Beam